

БЛОЧНАЯ МОДЕЛЬ

Преподаватель:
Пьянкова Марина
Анатольевна

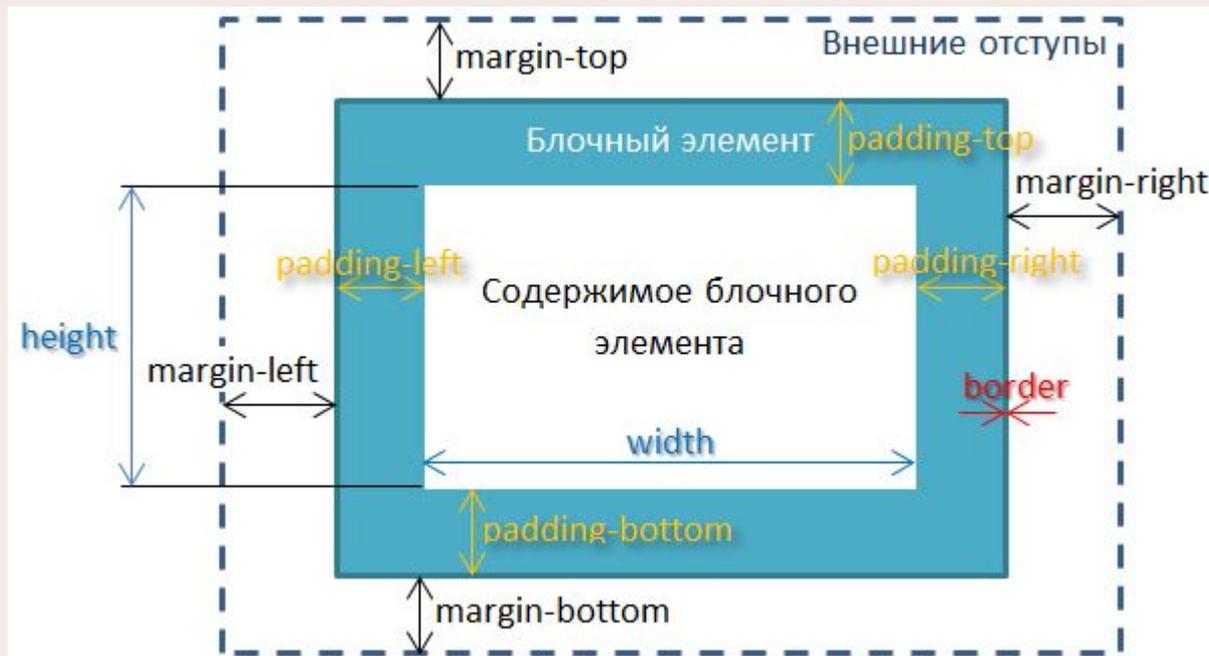
Блочная модель

1

В HTML все элементы можно представить в виде прямоугольных блоков. Для отображения блока, браузер вычисляет размеры его содержимого и применяет к блоку набор типовых свойств.

2

Таким образом, блочная модель является первым шагом к пониманию, как отображаются элементы в браузере. Знание блочной модели помогает строить отдельные блоки и компоновать их в веб-страницы, что и является основой вёрстки.



Тег <div>

! Тег **<div>** в HTML является блочным элементом верстки. Используется как контейнер для содержимого.

```
<div>  
    Содержимое  
</div>
```

! С помощью тега **<div>** обычно и создается внутренняя структура страницы. Тег **<div>** может содержать вложенные **<div>** элементы в неограниченном количестве.

! Для определения свойств блоков **div** используют каскадные таблицы стилей (CSS).

Свойство `box-sizing`



Свойство **`box-sizing`** определяет, как вычисляется ширина и высота элемента: должны ли они включать отступы и границы, или нет.
Синтаксис: **`box-sizing: content-box | border-box | initial | inherit;`**

```
box-sizing: content-box;  
width: 100%;
```



```
box-sizing: content-box;  
width: 100%;  
border: solid #5B6DCD 10px;  
padding: 5px;
```



```
box-sizing: border-box;  
width: 100%;  
border: solid #5B6DCD 10px;  
padding: 5px;
```



HTML5 - семантические теги



Список основных семантических тегов:

- `<header>`
- `<nav>`
- `<aside>`
- `<section>`
- `<main>`
- `<footer>`

- `<article>`

- `<figure>`
- `<figcaption>`

- `<details>`
- `<summary>`

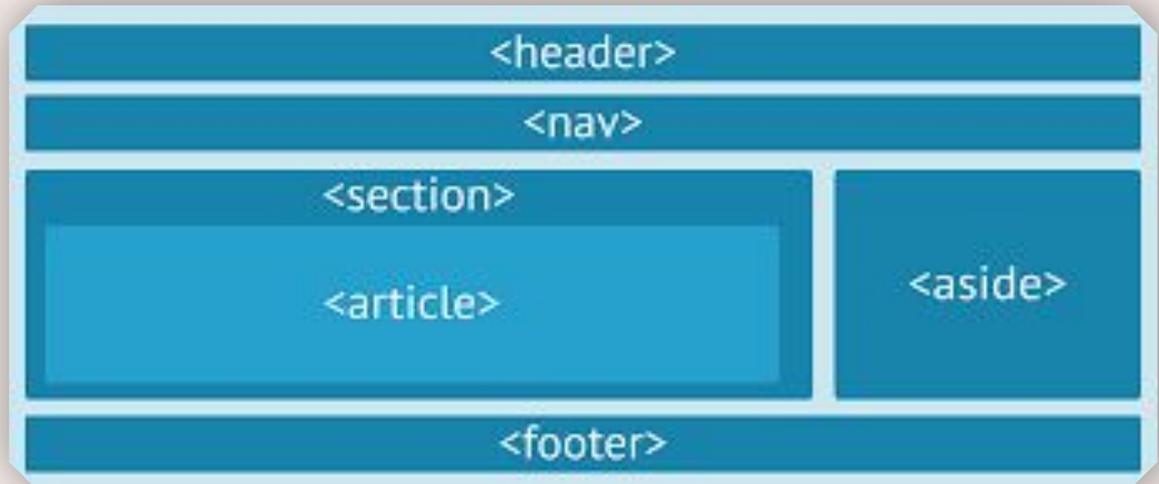
- `<mark>`
- `<time>`



Семантические элементы четко описывают, что они означают, как браузеру, так и веб-разработчику.

В качестве примера **не семантических элементов** можно привести теги `<div>` и ``. Они ничего не говорят о характере их контента.

Примеры **семантических элементов**: `<form>`, `<table>` и `<article>`. Они четко описывают, какого характера контент они содержат.



HTML5 - семантические теги



1 группа – внешний шаблон страницы:

Элемент **<header>** предназначен для определения заголовочного блока или "шапки" документа или раздела.

Элемент **<header>** следует использовать как контейнер для вводной информации. В одном документе разрешается определять несколько элементов **<header>**.

```
<article>
  <header>
    <h1>Что делает Всемирный фонд дикой природы (ВФП)?</h1>
    <p>Цель ВФП:</p>
  </header>
  <p>Задача Всемирного фонда дикой природы остановить деградацию окружающей среды на нашей планете и построить будущее, в котором человечество будет жить в гармонии с дикой природой.</p>
</article>
```

Элемент **<nav>** определяет набор ссылок навигации.

Обратите внимание, что НЕ ВСЕ ссылки в документе следует размещать внутри элемента **<nav>**. Элемент **<nav>** предназначен только для основного блока навигационных ссылок.

```
<nav>
  <a href='/html/'>HTML</a> |
  <a href='/css/'>CSS</a> |
  <a href='/js/'>JavaScript</a> |
  <a href='/jquery/'>jQuery</a>
</nav>
```

[HTML](#) | [CSS](#) | [JavaScript](#) | [jQuery](#)

HTML5 - семантические теги



1 группа – внешний шаблон страницы:

Элемент **<aside>** определяет некий контент, находящийся в стороне от контента, внутри которого он расположен (как боковой блок страницы, "сайдбар"). Контент внутри элемента **<aside>** должен соотноситься с окружающим контентом.

```
<p>Этим летом я с семьей посетил EPCOT центр.</p>
```

```
<aside>
```

```
  <h4>EPCOT центр</h4>
```

```
  <p> EPCOT центр – это тематический парк в развлекательном комплексе Уолта Диснея во Флориде.</p>
```

```
</aside>
```

Тег **<section>** определяет раздел документа, вроде блока новостей, глав текста, вкладок в диалоговом окне и т.п.

По умолчанию рассматривается браузером как блочный элемент. Допускается вкладывать в элемент **<section>** другие элементы **<section>**.

```
<section>
```

```
  <h1>Заголовок раздела</h1>
```

```
  <p>Текстовая информация. Lorem Ipsum dolor set amet....</p>
```

```
</section>
```

HTML5 - семантические теги



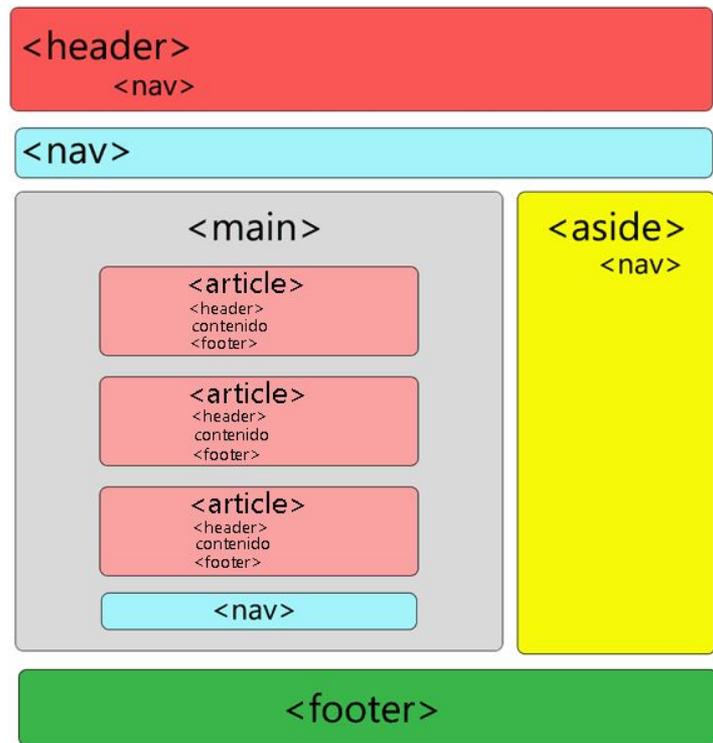
1 группа – внешний шаблон страницы:

Тег `<main>` определяет основное содержимое документа. Содержимое внутри элемента `<main>` должно быть уникальным для данного документа. Здесь не должно быть ничего, что повторяется где-либо, вроде сайдбаров, навигационных ссылок, логотипов, поисковых форм и т.п. В документе может быть только один элемент `<main>`, и он **не должен** быть **вложен** в элементы `<article>`, `<aside>`, `<footer>`, `<header>` или `<nav>`.

Элемент `<footer>` предназначен для определения "подвала" документа или раздела и должен содержать информацию о содержащем его элементе.

Обычно в "подвале" размещают информацию об авторе, ссылки на условия использования текста, информация об авторских правах, контактные данные и т.п.

В одном документе разрешается определять несколько элементов `<footer>`.



```
<footer>
  <p>Автор И.И.Иванов</p>
  <p>Контактная информация: <a href="mailto:someone@example.com">someone@example.com</a>.</p>
</footer>
```

HTML5 - семантические теги



2 группа – внешний шаблон страницы:

Элемент `<article>` определяет независимый, самодостаточный контент. Контент, помещенный в этот элемент, должен иметь смысл сам по себе, т. е. он должен быть понятен в отрыве от остальных частей веб-сайта.

В качестве примеров использования элемента `<article>` могут выступать:

- Публикация на форуме
- Публикация в блоге
- Газетная статья

```
<article>  
  <h1>Что делает Всемирный фонд дикой природы?</h1>  
  <p>Задача Всемирного фонда дикой природы остановить деградацию окружающей среды на нашей планете и построить будущее, в котором человечество будет жить в гармонии с дикой природой.</p>  
</article>
```

HTML5 - семантические теги

! 3 группа – элементы `<figure>` и `<figcaption>`:

Назначение элемента `<figcaption>` - добавление визуального пояснения к изображению.

В HTML5 изображение и пояснение к нему может быть сгруппировано в элементе `<figure>`:

```
<!DOCTYPE html>
<html>
<body>

<figure>
  
  <figcaption>Fig.1 - A view of the pulpit rock in Norway.</figcaption>
</figure>

</body>
</html>
```



Fig.1 - A view of the pulpit rock in Norway.

Элемент `` определяет изображение, а элемент `<figcaption>` пояснение к нему.

HTML5 - семантические теги

! 4 группа – элементы `<details>` и `<summary>`:

Тег `<details>` определяет дополнительную информацию, которую пользователь может по желанию открыть или скрыть. Внутри тега `<details>` можно размещать любой контент.

Содержимое элемента `<details>` должно быть скрыто до тех пор, пока не будет установлен атрибут `open`.

Для определения видимого заголовка дополнительной информации в теге `<details>` используется элемент `<summary>`. На заголовок можно нажимать. Это **откроет/закроет** дополнительную информацию.

```
<!DOCTYPE html>
<html>
<body>

<details>
  <summary>Copyright 1999-2019.</summary>
  <p> - by Refsnes Data. All Rights Reserved.</p>
  <p>All content and graphics on this web site are the property<br>
of the company Refsnes Data.</p>
</details>

<p><b>Note:</b> The details tag is not supported in Internet Explorer.
</p>

</body>
</html>
```

▶ Copyright 1999-2019.

Note: The details tag is not supported in Internet Explorer.



▼ Copyright 1999-2019.

- by Refsnes Data. All Rights Reserved.

All content and graphics on this web site are the property of the company Refsnes Data.

Note: The details tag is not supported in Internet Explorer.

HTML5 - семантические теги



5 группа – элементы `<mark>` и `<time>`:

Тег `<mark>` определяет маркированный/выделенный текст.

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

Do not forget to buy **milk** today.

```
<p><strong>Note:</strong> The mark tag is not supported<br>in Internet Explorer 8 and earlier versions.</p>
```

Note: The mark tag is not supported in Internet Explorer 8 and earlier versions.

Тег `<time>` определяет читаемую человеком дату и время.

Этот элемент также может использоваться для **кодирования дат и времени** в машинно-читаемом виде, чтобы агенты пользователей могли предлагать добавлять напоминания о днях рождения или запланированные события в календарь пользователя, а поисковые системы могут создавать более умные результаты поиска.

```
<!DOCTYPE html>
<html>
<body>

<p>We open at <time>10:00</time> every morning.</p>

<p>I have a date on <time datetime="2019-02-14 20:00">Valentines
day</time>.</p>

</body>
</html>
```

We open at 10:00 every morning.

I have a date on Valentines day.

Свойства блока



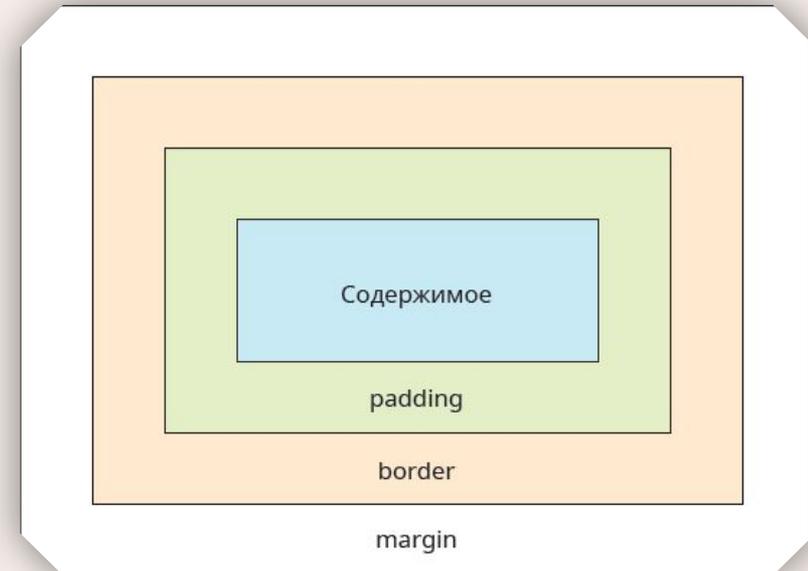
Как мы уже знаем, любой элемент на веб-странице представляет собой прямоугольный **блок**, который состоит из набора свойств.



Основой блока выступает его содержимое, **ширина** которого задается свойством **width**, а **высота** через **height**;



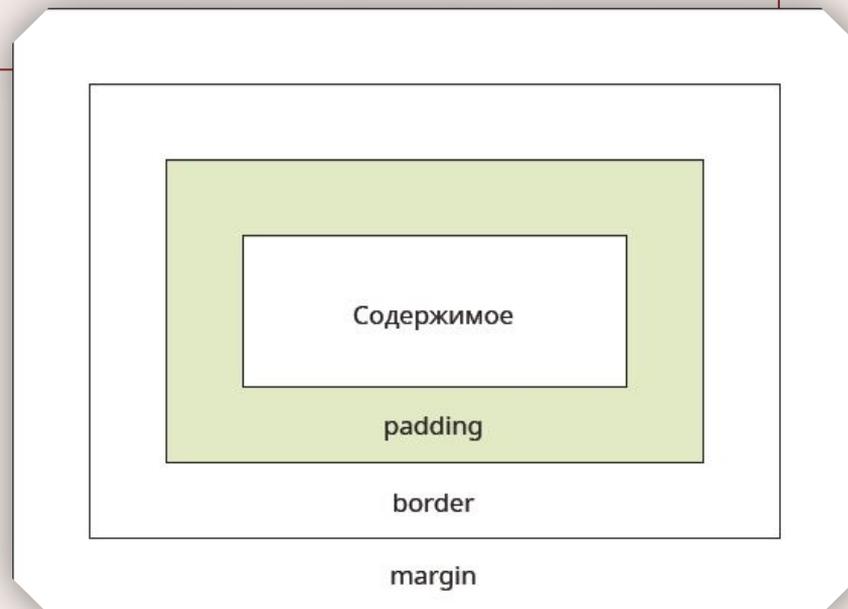
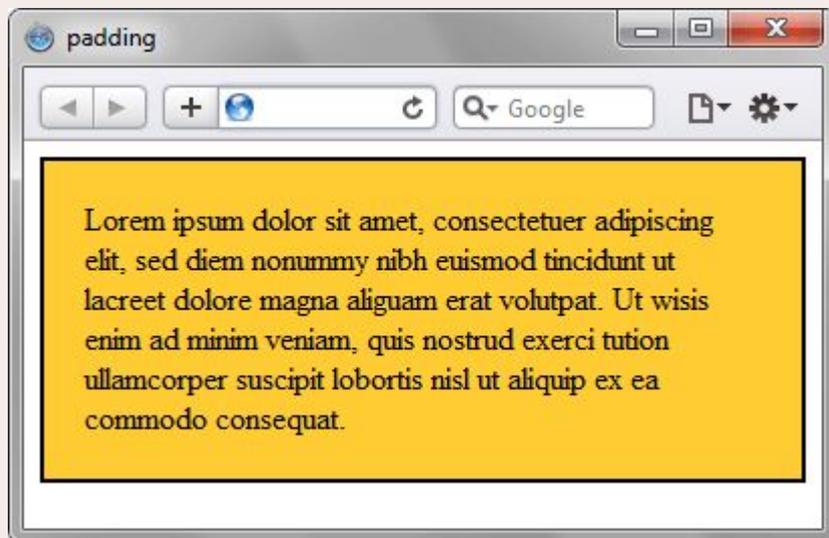
Вокруг содержимого идут поля (**padding** – **внутренние отступы**), они создают пустое пространство от содержимого до внутреннего края границ, затем идут границы (**border**), а вокруг границ внешние отступы (**margin**) — невидимое пустое пространство от внешнего края границ. Порядок этих свойств в блоке чётко определён.



Свойство Padding

! Свойство **padding** задаёт расстояние от внутреннего края границы или края блока до воображаемого прямоугольника, ограничивающего содержимое блока.

! Основное предназначение свойства **padding** - создать пустое пространство вокруг содержимого элемента, например текста, чтобы он не прилегал плотно ко краю элемента. Использование **padding** повышает читабельность текста и улучшает внешний вид страницы.



Формат записи Padding



Padding достаточно универсальное свойство и у него может быть указано от одного до четырех значений. Это нужно, чтобы одновременно задавать поля на разных сторонах элемента.



При указании **одного значения** данного свойства определяются поля для всех сторон блока.

```
padding: 10px; /* со всех сторон отступ 10px */
```



Два значения: первое определяет поля сверху и снизу для элемента, второе значение слева и справа для элемента.

```
padding: 10px 20px; /* сверху и снизу отступ 10px */  
/* слева и справа отступ 20px */
```



Три значения: первое задаст поле сверху для элемента, второе одновременно слева и справа, а третье снизу.

```
padding: 10px 3% 20px; /* сверху отступ 10px */  
/* слева и справа отступ 3% */  
/* снизу отступ 20px */
```

Формат записи Padding



Четыре значения: первое определяет поле сверху, второе справа, третье снизу, четвёртое слева.

```
padding: 1em 3px 30px 5px; /* сверху отступ 1em */
                          /* справа отступ 3px */
                          /* снизу отступ 30px */
                          /* слева отступ 5px */
```



Кроме того, есть свойства для каждой стороны отдельно - **padding-top**, **padding-right**, **padding-bottom** и **padding-left**.



Для информации!

Тег `<!-- -->` используется для написания комментария в коде HTML документа. HTML комментарии призваны улучшить читабельность кода. В комментариях обычно указывается объяснение участка кода, что упрощает процесс редактирования

Свойство text-align

1

Часто используемые значения свойства text-align:

Left: As most young candidates for the pains and penalties of whaling stop at this same New Bedford, thence to embark on their voyage, it may as well be related that I, for one, had no idea of so doing. For my mind was made up to sail in no other than a Nantucket craft, because there was a fine, boisterous something about everything connected with that famous old island, which amazingly pleased me. Besides though New Bedford has of late been gradually monopolizing the business of whaling, and though in this matter poor old Nantucket is now much behind her, yet Nantucket was her great original- the Tyre of this Carthage;- the place where the first dead American whale was stranded.

Right: Where else but from Nantucket did those aboriginal whalemens, the Red-Men, first sally out in canoes to give chase to the Leviathan? And where but from Nantucket, too, did that first adventurous little sloop put forth, partly laden with imported cobblestones- so goes the story- to throw at the whales, in order to discover when they were nigh enough to risk a harpoon from the bowsprit?

Center: Now having a night, a day, and still another night following before me in New Bedford, ere could embark for my destined port, it became a matter of concernment where I was to eat and sleep meanwhile. It was a very dubious-looking, nay, a very dark and dismal night, biting cold and cheerless. I knew no one in the place. With anxious grapnels I had sounded my pocket, and only brought up a few pieces of silver

Justify: So, wherever you go, Ishmael, said I to myself, as I stood in the middle of a dreary street shouldering my bag, and comparing the towards the north with the darkness towards the south- wherever in your wisdom you may conclude to lodge for the night, my dear Ishmael, be sure to inquire the price, and don't be too particular.

```
.left {
    text-align: left;
    border: 1px dotted black;
    width: 50%;
}
.right {
    text-align: right;
    border: 1px dotted black;
    width: 50%;
}
.center {
    text-align: center;
    border: 1px dotted black;
    width: 50%;
}
.justify {
    text-align: justify;
    border: 1px dotted black;
    width: 50%;
}
```

Свойство text-decoration

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-decoration: overline;
}
h2 {
  text-decoration: line-through;
}
h3 {
  text-decoration: underline;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>

</body>
</html>
```

This is heading 1

~~This is heading 2~~

This is heading 3

A link with no underline: html5css.ru

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
  text-decoration: none;
}
</style>
</head>
<body>

<p>A link with no underline:
<a href="https://html5css.ru">html5css.ru</a></p>

</body>
</html>
```

Свойство text-transform

```
<!DOCTYPE html>
<html>
<head>
<style>
p.uppercase {
  text-transform: uppercase;
}

p.lowercase {
  text-transform: lowercase;
}

p.capitalize {
  text-transform: capitalize;
}
</style>
</head>
<body>

<p class="uppercase">This is some text.</p>
<p class="lowercase">This is some text.</p>
<p class="capitalize">This is some text.</p>

</body>
</html>
```

THIS IS SOME TEXT.

this is some text.

This Is Some Text.

Свойство `text-indent`

1

Свойство `text-indent` используется для указания отступа первой строки текста:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-indent: 50px;
}
</style>
</head>
<body>

<p>In my younger and more vulnerable years
my father gave me some advice that I've
been turning over in my mind ever since.</p>

</body>
</html>
```

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.

Свойство border

1

Часто используемые значения свойства **border** и примеры:

```
border: 5px solid;
```

This is a box with a border around it.

```
border: dashed red;
```

This is a box with a border around it.

```
border: thick double #32a1ce;
```

This is a box with a border around it.

```
border: 4mm ridge rgba(170, 50, 220, .6);
```

This is a box with a border around it.

Вид рамки

1

Вид рамки в зависимости от стиля и толщины границы элемента

1 пиксел	3 пиксела	5 пикселей	7 пикселей
 dotted	 dotted	 dotted	 dotted
 dashed	 dashed	 dashed	 dashed
 solid	 solid	 solid	 solid
 double	 double	 double	 double
 groove	 groove	 groove	 groove
 ridge	 ridge	 ridge	 ridge
 inset	 inset	 inset	 inset
 outset	 outset	 outset	 outset

Свойство Margin



Свойство **margin**, в основном, используется для создания внешних вертикальных и горизонтальных отступов между элементами.



Аналогично другим блочным свойствам есть свойства для каждой стороны - **margin-top**, **margin-right**, **margin-bottom** и **margin-left**.



Свойство Margin



Примеры использования свойства **margin**.

```
margin: 10px;                /* все стороны: отступ 10px */

---

margin: 1.6em 20px;         /* верх и низ: отступ 1.6em */  
                           /* право и лево: отступ 20px */

---

margin: 10px 3% 1em;       /* верх: отступ 10px */  
                           /* право и лево: отступ 3% */  
                           /* низ: отступ 1em */

---

margin: 10px 3px 30px 5px; /* верх: отступ 10px */  
                           /* право: отступ 3px */  
                           /* низ: отступ 30px */  
                           /* лево: отступ 5px */

---

margin: 2em auto;          /* верх и низ: отступ 2em */  
                           /* блок отцентрирован горизонтально */

---

margin: auto;              /* верх и низ: отступ 0 */  
                           /* блок отцентрирован горизонтально */
```

Margin auto

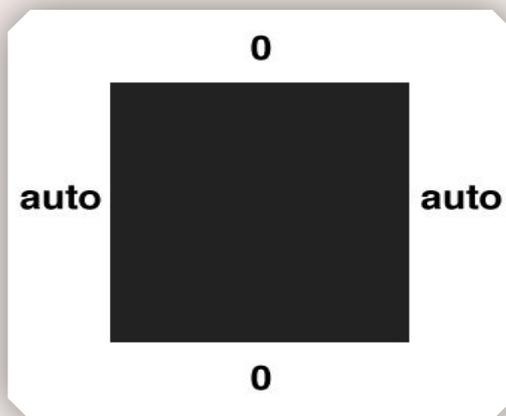
!

В отличие от **padding**, свойство **margin** может быть отрицательным (например: `-10px`), а также принимать значение **auto**. По вертикали значения **margin** не складываются, выбирается максимальное значение, а по горизонтали происходит сложение.

!

Если использовать значение **auto** для отступов слева и справа, а также задать ширину элемента через **width**, то свойство **margin** позволяет выравнивать элемент по центру горизонтали.

```
margin: 0 auto;
```



Обнуление `margin`



Для некоторых элементов, браузер добавляет **margin** автоматически. Вот несколько примеров:

- для `<body>` по умолчанию **margin** задан как **8px**;
- для списков ``, ``, `<dl>` **margin-top** и **margin-bottom** заданы как **1em (16px)**;



Эти **margin**, согласно макету, не всегда нужны, поэтому их можно обнулить, иными словами, задать для **margin** нулевое значение.

Аналогично можно сделать и со свойством **padding**.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

Позиционирование элементов



Позиционированием называется положение элемента в системе координат. Различают несколько типов позиционирования: нормальное, относительное, абсолютное, фиксированное и приклеенное. В зависимости от типа, который устанавливается через свойство **position**, меняется и система координат.



Благодаря комбинации свойств **position**, **left**, **top**, **right** и **bottom** можно отображать элемент в точке с заданными координатами, фиксировать его в указанном месте, определять положение одного элемента относительно другого, управлять размерами и др.

Свойство Position



Для изменения типа позиционирования применяется свойство **position**. Само по себе это свойство используется крайне редко и как правило идет в комбинации со свойствами **left**, **right**, **top**, **bottom**, которые определяют, соответственно, положение элемента слева, справа, сверху и снизу.

```
position: absolute;  
left: 10px;  
top: 10px;
```



Свойство **position** не только устанавливает тип позиционирования, но и меняет поведение элемента, благодаря чему расширяется арсенал приемов верстки.



Особенности приоритетов!

При указании свойств **left** и **right**, приоритет будет иметь **left**, а **right** – не работает. При указании **top** и **bottom**, приоритет будет иметь **top**.

Свойство Position

! У свойства **position** есть пять значений:

static

нормальное или статическое позиционирование, при этом элементы отображаются на веб-странице в том порядке, в каком они идут в исходном коде HTML сверху вниз. **static** - это значение по умолчанию для **position**. Свойства **left**, **top**, **right**, **bottom** и **z-index** не применяются к данному элементу.

```
position: static;
```

In this demo you can control the position property for the yellow box.

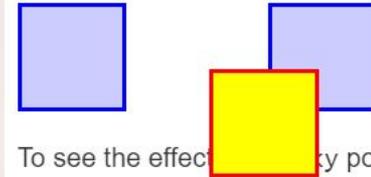


relative

относительное позиционирование. Изменяет положение элемента от его исходного расположения на основе значений **left**, **top**, **right**, **bottom**. Его место, которое он занимал по структуре макета страницы остается и не схлопывается. Положительное значение **left** сдвигает вправо от левой границы элемента, отрицательное - сдвигает влево. Положительное значение **top** сдвигает элемент вниз, отрицательное - сдвигает вверх. Работает свойство **z-index**.

```
position: relative;  
top: 40px; left: 40px;
```

In the property



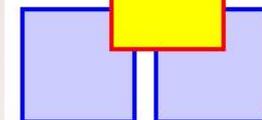
To see the effect of sticky positioning, select the position: sticky option and scroll this container.

absolute

абсолютное позиционирование. Элемент при этом не существует в потоке документа (его место схлопывается), его положение задаётся относительно его ближайшего относительно позиционированного предка. Если такого нет, относительно **body**.

```
position: absolute;  
top: 40px; left: 40px;
```

In the property



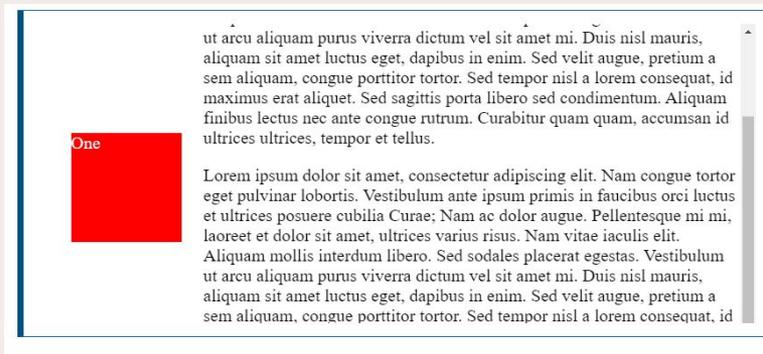
on yellow box.

Свойство Position

fixed

фиксированное позиционирование. По своему действию похоже на абсолютное позиционирование, но в отличие от него привязывается к указанной свойстами **left**, **top**, **right** и **bottom** точке на экране и не меняет своего положения при прокрутке веб-страницы.

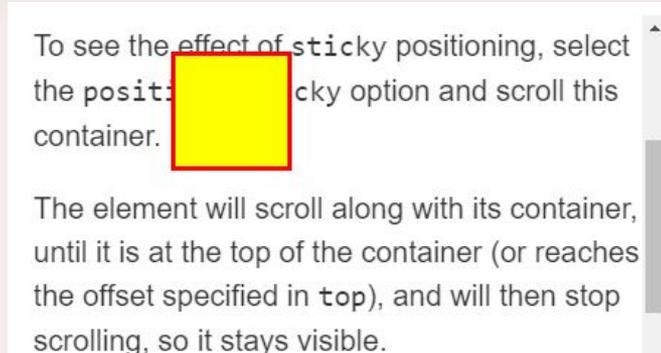
```
position: fixed;  
top: 80px; left: 10px;
```



sticky

липкое позиционирование. Обычно применяется для фиксации заголовка на одном месте, пока содержимое, к которому относится заголовок, прокручивается на странице.

```
position: -webkit-sticky;  
position: sticky;  
top: 20px;
```



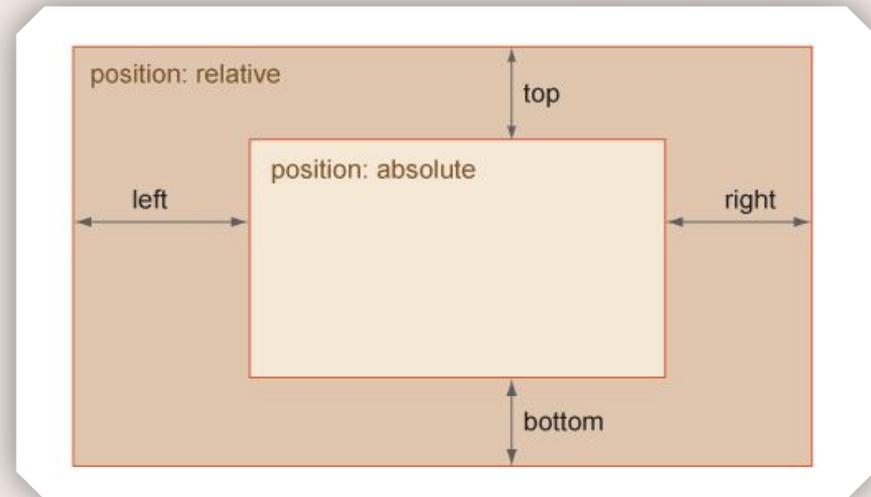
Relative and Absolute



Обычно, относительное позиционирование само по себе применяется не часто, поскольку для смещения элемента от его исходного положения есть ряд других свойств, к примеру, **margin** или **transform**. Но комбинация разных типов позиционирования для вложенных элементов является одним из удобных и практичных приемов верстки.



Если для родительского элемента задать **position: relative**, а для дочернего **position: absolute**, то произойдёт смена системы координат и положение дочернего элемента при этом указывается относительно его родителя, а не относительно **body**. Отсчёт координат ведётся от внутреннего края границы, значения **padding** не учитываются.



Position: fixed



Фиксированное позиционирование элемента задается значением **fixed** свойства **position** и по своему действию похоже на абсолютное позиционирование. Но, в отличие от него, привязывается к указанной свойстами **left**, **top**, **right** и **bottom** точке на экране и не меняет своего положения при прокрутке веб-страницы.



Особенности!

- При прокрутке веб-страницы элемент остаётся на своём исходном месте.
- Положение элемента всегда отсчитывается от области просмотра браузера, независимо от позиционирования родителя.
- При выходе элемента за пределы видимой области справа или снизу от нее, не возникает полос прокрутки и элемент просто «обрезается».

В остальном, значение **fixed** похоже на **absolute**. Применяется такой тип позиционирования для создания меню, вкладок, заголовков, в общем, любых элементов, которые должны быть закреплены на веб-странице и всегда видны посетителю.

```
.box {  
  background: yellow;  
  position: fixed;  
  left: 50%;  
  top: 50%;  
}
```

Lorem ipsum dolor sit amet
consectetur, adipisicing elit.
Obcaecati maxime fuga
quibusdam aperiam,
accusantium tempora nostrum
eveniet e **Fixed block!** sentium
perferendis panatur molestiae
suscipit debitis iure saepe modi
rerum quae.
Lorem ipsum dolor sit amet
consectetur, adipisicing elit.
Obcaecati maxime fuga

Float



Иногда возможностей свойства **position** может быть недостаточно или они будут не очень удобны в использовании. Например, в задаче, где нужно разместить несколько блоков горизонтально. В таком случае можно воспользоваться свойством **float**.



Свойство **float** позволяет сделать элемент обтекаемым. Цель обтекания элемента состоит в том, чтобы сдвинуть его к одной стороне и сделать так, чтобы текст обтекал вокруг элемента.

Float



Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugiat aspernatur fuga et aliquam ipsa optio, esse nobis saepe, pariatur iste quibusdam animi ea. Quibusdam mollitia aliquam voluptatibus asperiores nemo eaque! Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut nesciunt deserunt dolore animi necessitatibus asperiores quis earum. Perferendis, animi, eligendi ipsam blanditiis provident possimus nobis officiis corrupti consequatur pariatur est. Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ipsam excepturi, eveniet sapiente nesciunt laboriosam, delectus architecto ipsa molestiae odio, ab, magnam veritatis et? Voluptatum officiis dolorum exercitationem fuga cum nemo. Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ipsam excepturi, eveniet sapiente nesciunt laboriosam, delectus architecto ipsa molestiae odio, ab, magnam veritatis et? Voluptatum officiis dolorum exercitationem

```
img {  
  float: left;  
}
```

!

Как вы можете видеть, изображение сдвинулось влево, а текст просто обтекает вокруг изображения:

- сперва текст сдвинулся вправо, рядом с изображением;
- затем, когда появилось пространство под изображением, текст заполнил это пространство.

Float



Свойство **float** является очень мощным инструментом, однако, неправильное его использование может нарушить структуру всей страницы и привести к неожиданным последствиям.



Так как **float** “ломает” поток элементов в документе, после его использования нужно вернуть все на свои места. Для этого можно использовать шаблонное правило в CSS, например:

```
.clearfix::after {  
  content: '';  
  clear: both;  
  display: block;  
}
```



Класс **.clearfix** использует свойство **clear**, которое очищает последствия свойства **float** и возвращает элементы в нормальное значение.

По необходимости, добавьте данный класс к любому элементу, после которого хотите очистить поток.